Rank3DGAN: Semantic Mesh Generation Using Relative Attributes

Yassir Saquil*, **Qun-Ce Xu***, **Yong-Liang Yang, and Peter Hall**
University of Bath, Bath, UK
{ys999, qx289, yy753, maspmh}@bath.ac.uk

Abstract

In this paper, we investigate a novel problem of using generative adversarial networks in the task of 3D shape generation according to semantic attributes. Recent works map 3D shapes into 2D parameter domain, which enables training Generative Adversarial Networks (GANs) for 3D shape generation task. We extend these architectures to the conditional setting, where we generate 3D shapes with respect to subjective attributes defined by the user. Given pairwise comparisons of 3D shapes, our model performs two tasks: it learns a generative model with a controlled latent space, and a ranking function for the 3D shapes based on their multi-chart representation in 2D. The capability of the model is demonstrated with experiments on HumanShape, Basel Face Model and reconstructed 3D CUB datasets. We also present various applications that benefit from our model, such as multi-attribute exploration, mesh editing, and mesh attribute transfer.

Introduction

Building a generative model for 3D shapes has a wide variety of applications in computer vision and graphics, in both research and industrial fields. For instance, we can design a system for a computer animator so that he can manipulate 3D shapes not only based on low-level attributes such as geometry, but also high-level semantic attributes such as expressions for 3D faces or girth for 3D human bodies.

However, learning a generative model for fine-grained quality 3D shapes remains a challenging problem. The current state-of-the-art methods rely on representing the 3D shape as a tensor data itself which begets many limitations. For volumetric grid (*i.e.,* voxel) representation (Choy et al. 2016; Wu et al. 2016), surface details such as smoothness and continuity are lost due to limited resolution of voxel tensors. In contrast, point cloud (Qi et al. 2017) representation is simple, but it lacks a regular structure to easily fit into neural networks and it does not model connectivity between points, making 3D surface reconstruction a non-trival task. These drawbacks hinder the extension of these methods to more sophisticated tasks such as, semantic mesh manipulation.

* Equal contribution

Recently, Hamu *et al.* (2018) proposed to represent 3D meshes using 2D multi-chart structure based on orbifold tutte embedding (Maron et al. 2017). The basic idea is to cut the mesh (or part of it) through landmark points on the surface, then map it to a chart structure in 2D, and finally sample a grid in 2D. This results in a tensor of image-like data, which is both smooth and bijective. Moreover, the convolution operation is well-defined and can be mapped back to the original surface. Hence GANs (Goodfellow et al. 2014) can be trained based on multi-chart structure to generate charts tensor, which implicitly represent a 3D mesh surface.

Inspired by the semantic image manipulation using RankCGAN (Saquil, Kim, and Hall 2018), our aim is to extend this work to 3D mesh surfaces using the multi-chart structure (Hamu et al. 2018), which enables us to provide a deep learning system that can generate 3D meshes while manipulating some semantic attributes defined by the user. The key characteristic of this approach is the addition of a ranking module that can order 3D shapes via their multi-chart representation. Such a system is particularly useful to digital artists and designers since they can set the semantic attributes using pairwise comparisons in order to perform personalized 3D shapes editing and exploration.

We evaluated our method on three types of datasets of human bodies (HumanShape (Pishchulin et al. 2017)), birds (reconstructed 3D shapes of CUB imagery data (Kanazawa et al. 2018)), and human faces (Basel Face Model (Paysan et al. 2009; Gerig et al. 2018)). Both quantitative and qualitative results show that: Our model can disentangle semantic attributes while ensuring a coherent variation of high-quality 3D meshes thanks to the usage of multi-chart structure in comparison to existing methods. Also, our model has important applications in mesh generation, editing, and transfer.

Our contributions are two-fold: 1) a novel conditional generative model that can generate quality 3D meshes with semantic attributes quantified by a ranking function; 2) an end-to-end training scheme that only requires pairwise comparison of meshes instead of global annotation of the dataset.

Related work

3D shape generative models Early works on 3D shape synthesis relied on probabilistic inference methods. (Kaloger-

akis et al. 2012) proposed a probabilistic generative model of component-based 3D shapes, while (Xue, Liu, and Tang 2012) reconstructed 3D geometry from 2D line drawings using MAP estimate. These models are generally adequate for a certain class of 3D shapes.

Recent works shifted towards using deep generative models for 3D data synthesis. These works can be categorized according to the learned 3D representation, which fall into three types: voxel, point clouds and depth map based methods. Unlike the component-based methods, the voxel-based methods do not rely on part labeling and learn voxel grids from a variety of inputs; a probabilistic latent space (Wu et al. 2016), single or multi-view images (Choy et al. 2016; Girdhar et al. 2016). However, generated 3D voxels are generally low resolution and memory costly.

Another line of work focused on generating 3D point clouds (Achlioptas et al. 2018). This formulation avoids resolution issue, but induces challenges to ensure the order and transformation invariance of the 3D points (Qi et al. 2017). Another limitation of point clouds is the lack of point connectivity, which complicates the task of reconstructing a mesh.

Finally, some works considered generating depth maps, which are used as an intermediate step along with an estimated silhouette or normal map to generate a 3D shape (Soltani et al. 2017).

3D mesh surface learning There has been an increasing interest to generalize deep learning operators to curved surface meshes. (Boscaini et al. 2016; Masci et al. 2015) formulated the convolution operation in the non-Euclidean domain as template matching with local patches in geodesic or diffusion system. (Monti et al. 2017) proposed a mixture model networks on graphs and surfaces and built parametric local patches instead of fixed ones. These methods were demonstrated in the tasks of shape correspondence and description.

In another spectrum, many works processed triangular meshes with NNs despite their irregular format. (Tan et al. 2018) proposed a VAE to encode meshes using rotation-invariant representation for shape embedding and synthesis.

In addition, few works (Chen and Zhang 2019; Mescheder et al. 2019) focused on representing the 3D surface as a decision boundary of a classifier learned using voxelized shapes. These implicit models allow generating high-quality meshes using isosurface extraction algorithms.

Surface parameterization for mapping 3D surfaces to a 2D domain is a well studied problem in computer graphics. (Sinha, Bai, and Ramani 2016) learned CNN models using geometry images (Gu, Gortler, and Hoppe 2002). However, geometry images are neither seamless nor unique, making the convolution operation not translation-invariant. (Maron et al. 2017) tackled these issues by parameterizing a 3D surface to a global seamless flat-torus, where the convolution is well defined. These methods are restricted to sphere-type shapes and are applied to shape classification, segmentation and landmark detection tasks.

For 3D mesh surface synthesis, (Sinha et al. 2017) learned to generate geometry images as a shape representation. On the other hand, AtlasNet (Groueix et al. 2018) used MLPs to learn multiple parameterizations that map 2D squares with latent shape features to the 3D surface. Meanwhile, based on a sparse set of surface landmarks, (Hamu et al. 2018) represented a 3D shape by a collection of conformal charts formed by flat-torus parameterization (Maron et al. 2017). This reduces mapping distortion and can be used for multiple tasks that requires mesh quality, notably mesh generation using GANs (Goodfellow et al. 2014). Our work is an extension of multi-chart 3D GAN (Hamu et al. 2018) in the conditional setting using relative attributes.

Relative attributes In early studies, binary attributes that indicate the presence or the absence of an attribute in data showed state-of-the-art performance in object recognition (Tao, Smeulders, and Chang 2015) and action recognition (Liu, Kuipers, and Savarese 2011).

However, a better representation of attributes is necessary if we want to quantify the emphasis of an attribute and compare it with others. For this reason, relative attributes (Parikh and Grauman 2011) tackled this issue by learning a global ranking function on data using constraints describing the relative emphasis of attributes (*e.g.,* pairwise comparisons of data). This approach is regarded as solving a learning-to-rank problem where a linear function is learned based on RankSVM (Joachims 2002). This problem can also be modeled by a non-linear function as in RankNet (Burges et al. 2005), where the ranker is a neural network trained using gradient descent methods. Furthermore, semi-supervised learning can as well learn user-defined ranking functions as demonstrated in criteria sliders (Tompkin et al. 2017).

While these algorithms focus on predicting attributes on existing data entries, another line of works were interested in manipulating data using semantic attributes. For 3D shape generation task, (Streuber et al. 2016) used semantic attributes as high-level word description to generate 3D human shape. (Chaudhuri et al. 2013) proposed an interactive part-based assembly method for 3D shape creation associated with semantic attributes. For 3D shape editing task, (Yümer et al. 2015) proposed a method where the user creates continuous geometric deformations using a set of semantic attributes.

For image generation task, (Yan et al. 2016) separated the background and foreground generation. (Kaneko, Hiramatsu, and Kashino 2018) proposed a decision tree latent controller for GAN to capture salient features of the generated images. Lastly, RankCGAN (Saquil, Kim, and Hall 2018) proposed an extension enabling to continuously synthesize new imagery data according to semantic attributes. Our work can also be treated as an extension of RankCGAN to 3D mesh synthesis with respect to subjective criteria.

Approach

In this section, we describe in details our Rank3DGAN model, which consists of a conditional GAN capable of generating 3D mesh representation while manipulating predefined semantic attributes. Firstly, we start with a presentation of the 2D CNN adaptation for 3D surfaces using mesh parameterization method, which describes the process of mapping 3D surface to 2D images (charts). Secondly, we highlight the

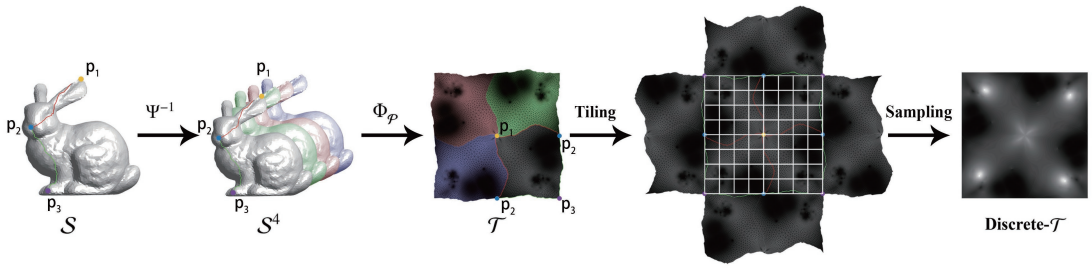

Figure 1: The process of computing a discretized chart from a 3D mesh, where $\mathcal{S}$ the original surface, $\mathcal{S}^4$ the 4-cover of the surface, $\mathcal{T}$ the 2D flat-torus, and discrete-$\mathcal{T}$ the obtained chart after sampling and tiling the torus to fit a square. Since the parameterization is bijective, A reconstruction of the original mesh can be performed from the resulting chart.

multi-chart structure as a representation of a 3D mesh in a tensor of charts, where each chart models a part of the mesh and the tensor covers the whole mesh with minimal distortion. We also describe multi-chart structure necessary geometric properties satisfied by the addition and modification of some CNN layers. Thirdly, we detail the loss functions of conditional GAN trained on this representation, which consists of a WGAN-GP loss for tensor of charts generation and a ranking loss for ensuring the ranking constraint on the semantic ordering of tensors of charts. Lastly, we show how we can extend Rank3DGAN to multiple attributes generation using a multi-label ranking loss and how Rank3DGAN can be used for mesh editing by training an additional encoder.

Seamless toric covers

Building a generative model for 3D meshes consists of learning a function $G : \mathbb{R}^d \to \mathcal{S}$ that maps a latent distribution in $\mathbb{R}^d$, usually a gaussian or uniform noted by p_z, to a complex distribution p_g in $\mathcal{S}$ that approximates the real distribution p_x of our training surfaces $\{S_i\}_{i\in I}$, where d is the dimension of the latent space and $\mathcal{S}$ is the surface space.

GANs (Goodfellow et al. 2014) are popular generative models for 2D images. But CNN architectures cannot be applied directly on data sampled from $\mathcal{S}$, which show the necessity of transferring the learning problem from surface to image space where the convolution operation is defined. For this reason, (Maron et al. 2017) suggest to transfer functions over the surface to a flat-torus, represented as a planar square $[-1, 1]^2$ and denoted as $\mathcal{T}$. The flat-torus is handy because we can discretize it to a $n \times n$ grid and apply standard 2D convolutions directly on the sampled grid.

Formally, in order to build a seamless mapping between $\mathcal{S}$ and $\mathcal{T}$ domains, an intermediate surface, $\mathcal{S}^4$, of 4-cover of $\mathcal{S}$ is created as follows; four copies of the surface are cut along the path $p_1 \to p_2 \to p_3$ designed by triplet of points $\mathcal{P} = \{p_1, p_2, p_3\} \subset \mathcal{S}$ to get disk-like surface; Then, the four surfaces are stitched to get a surface torus, $\mathcal{S}^4$; Afterwards, the conformal mapping $\Phi_{\mathcal{P}} : \mathcal{S}^4 \to \mathcal{T}$ is computed using the parameterization method (Maron et al. 2017); Lastly, the flat torus is tiled to cover the discretization grid resulting into a sampled chart as shown in the Figure 1.

In the case of $\mathcal{S}$ being a disk-like surface, e.g. 3D face meshes, a surface cut is not needed and a quadriplet of points $\mathcal{P} = \{p_1, p_2, p_3, p_4\} \in \partial\mathcal{S}$ is designated, with $\partial\mathcal{S}$ the surface boundary, in order to stitch the four surfaces to get a torus. $\text{push}_{\mathcal{P}}(x) = x \circ \Psi \circ \Phi_{\mathcal{P}}^{-1} \in \mathbb{R}^{c\times n\times n}$ is the obtained conformal chart after transferring a function $x \in \mathcal{F}(\mathcal{S}, \mathbb{R}^c)$ over the surface to a flat-torus using $\Phi_{\mathcal{P}}$, where $\Psi : \mathcal{S}^4 \to \mathcal{S}$ is the projection of 4-cover surface to the original one. A conformal chart produces an area scaling with respect to the selected triplet $\mathcal{P}$, thus a careful choice of multiple triplets can ensure a global coverage of the surface with a minimum distortion.

Multi-chart structure

The multi-chart structure (Hamu et al. 2018) is a set of charts that globally cover the surface of mesh. Formally, it is a tuple (P, F), where $P \in \mathbb{R}^{n\times 3}$ is the set of landmarks and F is the set of landmark triplets, such that each triplet $\mathcal{P} = \{p_i, p_j, p_k\} \in F$ represents a chart and every mesh S_i consists of $|F|$ charts.

Besides the coverage property, the multi-chart structure should ensure the scale-translation (s-t) rigidity (Hamu et al. 2018) that allows unique recovering the original scale and mean of the charts after the normalization applied while training a network.

Recall that our purpose is to learn a generative model for 3D meshes, which leads us to consider the coordinates $X = (x, y, z)$ as functions over the mesh $X \in \mathcal{F}(\mathcal{S}, \mathbb{R}^3)$ to be transferred using the multi-chart structure (P, F) as follows:

$$\text{push}_{\mathcal{P}}(X) = X \circ \Psi \circ \Phi_{\mathcal{P}}^{-1} \in \mathbb{R}^{3\times n\times n}$$

where $\text{push}_{\mathcal{P}}(X)$ is the obtained chart with spatial dimension $n \times n$ using the triplet $\mathcal{P} \in F$. By concatenating the charts obtained from using all triplet in F, we get the final multi-chart tensor representing the whole mesh defined by: $C \in \mathbb{R}^{3|F|\times n\times n}$. Since every 3 charts in C represent a different part in the mesh, we need to normalize them for an optimal learning process. In the end, the scale-translation rigidity property ensures a unique reconstruction.

Thanks to the multi-chart structure, learning a generative model $G : \mathbb{R}^d \to \mathbb{R}^{3|F|\times n\times n}$ for 3D meshes can be performed in the image space using GANs (Goodfellow et al. 2014) with considerations for the geometric setting described below:

- Standard convolution and deconvolution are substituted by cyclic-padding ones (Maron et al. 2017; Hamu et al. 2018) to take into account the invariance to torus symmetry.
- Projection layer is incorporated in the generator to satisfy the invariance of $\mathcal{S}^4$ symmetries.

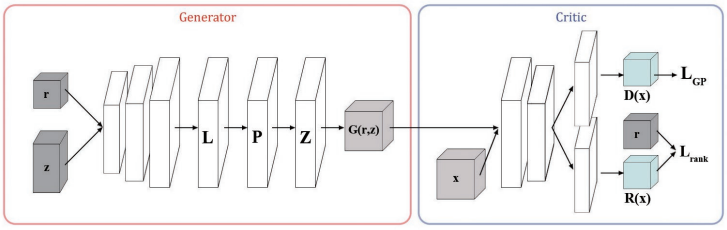

Figure 2: Rank3DGAN network: The dark grey represents the latent variables, light grey represents the chart images, light blue indicates the outputs of the critic, and the dark blue indicates their loss functions. The layers L, P, Z represent the landmark consistency, projection and zero-mean layers respectively.

- Landmark consistency layer (Hamu et al. 2018) is incorporated to force the generated normalized charts $G(z) \in \mathbb{R}^{3|F|\times n\times n}, z \in \mathbb{R}^d$ to satisfy the s-t rigidity property.
- Zero-mean layer is proposed to reduce the mean of every generated chart after using the landmark consistency layer.

The final step following learning a generative model G is to reconstruct the 3D mesh from the generated charts $C = G(z) \in \mathbb{R}^{3|F|\times n\times n}$ using template fitting (Hamu et al. 2018).

Since C is output of the landmark consistency layer, we can uniquely recover the scale and mean of charts resulting in charts denoted by $\hat{C}$. Afterwards, a template mesh S^t is used to obtain the generated mesh by setting the mesh vertices location using the multi-chart structure (P, F) and the generated charts $\hat{C}$ as follows:

$$v = \frac{\sum_{\mathcal{P}\in F} \tau_{\mathcal{P}}(v)\, \hat{C}(\Phi_{\mathcal{P}}(v))}{\sum_{\mathcal{P}\in F} \tau_{\mathcal{P}}(v)}$$

where $\tau_{\mathcal{P}}(v)$ is the inverse area scale of the 1-ring of vertex v exhibited by flat-torus $\Phi_{\mathcal{P}}(S^t)$ of the template mesh and $\hat{C}(\Phi_{\mathcal{P}}(v))$ is the image of the flat-torus vertex v under the learned charts in $\hat{C}$ associated with triplet $\mathcal{P}$ computed using bilinear interpolation of the grid cells in these learned charts.

Semantic mesh generation

We recall that Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) is defined as a minmax game between two networks; A generator G that maps a latent variable $z \sim p_z$ to generated sample $G(z)$, and a discriminator D that classifies an input sample as either a real or generated sample. Due to the training instability with the former loss, WGAN (Arjovsky, Chintala, and Bottou 2017) proposes using Wasserstein distance defined as the minimum cost of transporting mass to transform the generated distribution into real distribution. The WGAN loss suggests clipping the D (called a *critic*) weights to enforce the Lipschitz constraint.

WGAN-GP (Gulrajani et al. 2017) proposes another alternative for setting the Lipschitz constraint, which consists of a penalty on the gradient norm, controlled with a hyperparameter $\lambda = 10$, for random samples $\hat{x} \sim p_{\hat{x}}$ with $p_{\hat{x}}$ a uniform sampling along straight lines between samples from real p_x and generated p_g distributions. The WGAN-GP (Gulrajani et al. 2017) optimizes the following loss:

$$\mathcal{L}_{GP} = \mathbb{E}_{\tilde{x}\sim p_g}[D(\tilde{x})] - \mathbb{E}_{x\sim p_x}[D(x)] + \lambda\, \mathbb{E}_{\hat{x}\sim p_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2]$$

In order to generate meshes controlled by one or more subjective attributes, we need to incorporate semantic ordering constraints in WGAN-GP loss. We follow (Saquil, Kim, and Hall 2018) in modelling the semantic constraints by a ranking function, noted R, to be incorporated in the training of critic and generator. Training a pairwise ranker R using CNNs was proposed in (Burges et al. 2005; Souri, Noury, and Adeli 2016), which consists of learning to classify a pair of inputs, $x^{(1)}$ and $x^{(2)}$ according to their ordering, $x^{(1)} \succ x^{(2)}$ or $x^{(1)} \prec x^{(2)}$. Formally, given a dataset $\{(x_i^{(1)}, x_i^{(2)}, y)\}_{i=1}^P$ of size P, such that $(x_i^{(1)}, x_i^{(2)})$ pair of images and $y_i \in \{0, 1\}$ a binary label indicating if $x_i^{(1)} \succ x_i^{(2)}$ or not, we define the ranking loss as follows:

$$\mathcal{L}_{rank}(x_i^{(1)},\ x_i^{(2)},\ y_i) = -y_i \log[\sigma(R(x_i^{(1)}) - R(x_i^{(2)}))] - (1 - y_i) \log[1 - \sigma(R(x_i^{(1)}) - R(x_i^{(2)}))]),$$

with $\sigma(x)$ the sigmoid function. The ranker R can be regarded as a function that outputs the ranking score of the input data.

With all necessary components presented, we propose an architecture, Rank3DGAN, that can generate meshes according to subjective attributes set by the user. The underlying mechanism is the addition of semantic ordering constraint, defined by a ranking function, to the generative model trained using multi-chart structure representing the 3D meshes.

A fundamental difference between our work and (Saquil, Kim, and Hall 2018) is in the application context, since we focus on manipulating 3D shapes via the multi-chart structure. We also opted for architecture changes. As illustrated in Figure 2, instead of having a separate network for the ranker as in (Saquil, Kim, and Hall 2018), we introduce the ranker in the critic network D as an auxiliary classifier (Odena, Olah, and Shlens 2017), so that the critic D has two outputs for adversarial and ranking losses respectively. The idea behind dwells in the fact that with a separate ranker architecture, an end-to-end training with critic and generator is not required. The ranker could be trained off-line and plugged while the generator is trained. While having the ranker as an auxiliary classifier ensures that the training is end-to-end and the ranker can benefit from the learned representation of the critic.

In the following subsections, we describe Rank3DGAN in the mesh generation task with respect to one attribute, then we generalize to multiple attributes. Finally, we introduce an inference network that estimates the latent variables of an input multi-chart tensor for the task of mesh editing.

Dataset	# meshes	attributes	# pairs	# lmks	# triplets
MPII Human Shape	15,000	weight, gender	5,000	21	16
CUB-200-2011	5,000	lifted wings	5,000	14	11
Basel Face Model 09	5,000	gender, height, weight, age	5,000	4	N/A
Basel Face Model 17	5,000	anger, disgust, fear, happy, sad, surprise	5,000	4	N/A

Table 1: We prepared data from four datasets for our experiments. 'lmks' stands for landmarks.

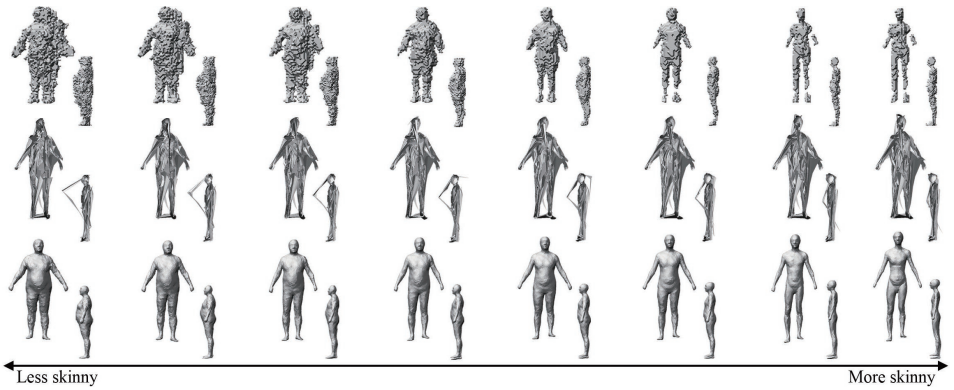

Figure 3: Generated interpolation of human shape with respect to *weight* attribute in the following methods: **Top:** Voxel 3D GAN. **Middle:** AtlasNet. **Bottom:** Our method Rank3DGAN.

One attribute generation As shown in Figure 2, the generator G outputs a multi-chart tensor $G(r, z)$ given two inputs, the noise latent vector $z \sim \mathcal{N}(0, 1)$ and the attribute latent variable $r \sim \mathcal{U}(-1, 1)$. The critic D takes a real $I = x$ or fake $I = G(r, z)$ multi-chart tensor and outputs its critic value $D(I)$ and ranking score $R(I)$. Given $\{(x_i^{(1)}, x_i^{(2)}, y_i)\}_{i=1}^{P}$ and $\{x_i\}_{i=1}^{N}$, mesh pairwise comparisons and mesh datasets represented by their multi-chart structure, with size of P and N respectively, we train our model in mini-batch setting of size B by defining the critic ($\mathcal{L}_D$) and generator ($\mathcal{L}_G$) loss functions as follows:

$$\mathcal{L}_D = \frac{1}{B}\sum_{i=1}^{B}[D(\tilde{x}_i) - D(x_i) + \lambda\,(\|\nabla_{\hat{x}_i} D(\hat{x}_i)\|_2 - 1)^2] + \frac{2}{B}\sum_{i=1}^{B/2}\mathcal{L}_{rank}(x_i^{(1)}, x_i^{(2)}, y_i),$$

$$\mathcal{L}_G = -\frac{1}{B}\sum_{i=1}^{B} D(\tilde{x}_i) + \nu\,\frac{2}{B}\sum_{i=1}^{B/2}\mathcal{L}_{rank}(\tilde{x}_i^{(1)}, \tilde{x}_i^{(2)}, [r_i^{(1)} > r_i^{(2)}]),$$

with $\tilde{x}_i = G(r_i, z_i)$, $\hat{x}_i = \alpha\tilde{x}_i + (1-\alpha)x_i$, for $\alpha \sim \mathcal{U}(0, 1)$, $\tilde{x}_i^{(1)} = G(r_i^{(1)}, z_i^{(1)})$, $\tilde{x}_i^{(2)} = G(r_i^{(2)}, z_i^{(2)})$. [.] is Iverson bracket. ν (hyperparameter) controls the ranker gradient magnitude in the generator update.

Multiple attributes generation Our method can be extended to the multiple attributes case where the attribute latent space is vector representing the set of controllable attributes and the critic network D outputs a vector of ranking scores with respect to each attribute. Formally, let $\{(x_i^{(1)}, x_i^{(2)}, \mathbf{y_i})\}_{i=1}^{P}$ mesh pairwise comparisons dataset with $\mathbf{y_i}$ a binary vector indicating whether $x_i^{(1)} \succ x_i^{(2)}$ or not with respect to all attributes A. The new ranking loss is defined as follows:

$$\mathcal{L}_{m-rank}(x_i^{(1)}, x_i^{(2)}, \mathbf{y_i}) = -\sum_{j=1}^{A} y_{ij}\log[\sigma(R_j(x_i^{(1)}) - R_j(x_i^{(2)}))] + (1 - y_{ij})\log[1 - \sigma(R_j(x_i^{(1)}) - R_j(x_i^{(2)}))])$$

with y_{ij}, $R_j(x_i^{(1)})$, $R_j(x_i^{(2)})$ the j-th elements in $\mathbf{y}_i$, $R(x_i^{(1)})$, $R(x_i^{(2)})$ vectors respectively. By substituting $\mathcal{L}_{rank}$ in Equations $\mathcal{L}_G$ and $\mathcal{L}_D$ with $\mathcal{L}_{m-rank}$, we obtain the generative model with multiple attributes.

Encoder for mesh editing The last component to add to our generative model, is an inference network for tasks that require estimating the latent variables of an input mesh such as mesh editing or attributes transfer. Following the previous work (Zhu et al. 2016) in image manipulation, we propose to train the encoder E on the real multi-chart tensors dataset $\{x_i\}_{i=1}^{N}$ in order to estimate the latent variables that minimise the reconstruction loss defined in the mini-batch setting as:

$$\mathcal{L}_E = \frac{1}{B}\sum_{i=1}^{B} \| G(E(x_i)) - x_i \|_2^2 .$$

Empirical results

In this section, we describe our experimental settings, show quantitative and qualitative results, and then demonstrate applications in mesh generation, mesh editing, and mesh attribute transfer.

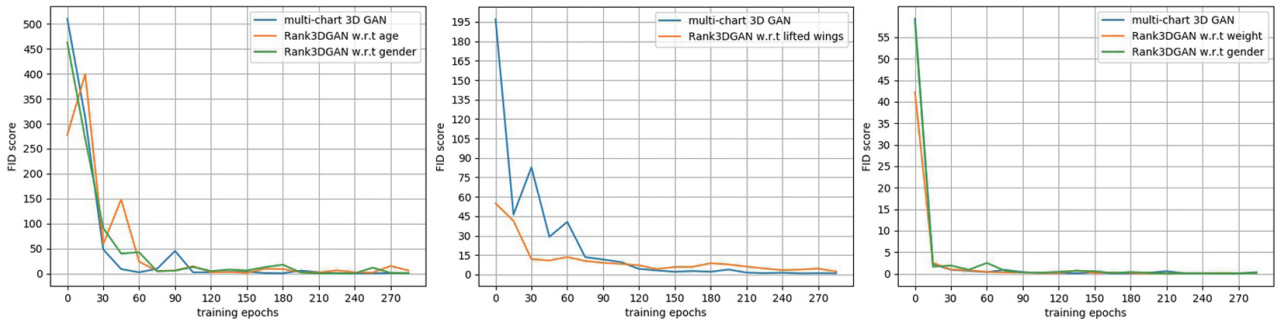

Figure 4: Mean-FID on face, bird and human datasets. **Left:** face dataset trained on *age* and *gender*. **Middle:** bird dataset trained on *lifted wings*. **Right:** human dataset trained on *gender* and *weight*.

Dataset	IoU			Chamfer			Normal Consistency		
	Rank3DGAN	AtlasNet	Voxel 3DGAN	Rank3DGAN	AtlasNet	Voxel 3DGAN	Rank3DGAN	AtlasNet	Voxel 3DGAN
Human	**0.8146**	-	0.6426	**0.0074**	0.0129	0.0084	**0.9561**	0.8116	0.8535
Bird	**0.7882**	-	0.4192	**0.0246**	0.0428	0.0684	**0.9362**	0.8612	0.7353
Face	-	-	-	0.0273	**0.0237**	0.0373	**0.9386**	0.9335	0.5577

Table 2: Quantitative comparison of our approach against the baselines in the mesh generation task.

Data preparation

We relied on four datasets for our experiments: 3D human shapes from MPII Human Shape (Pishchulin et al. 2017), 3D bird meshes reconstructed from CUB image dataset (Kanazawa et al. 2018), and 3D faces from Basel Face Model 2009 (Paysan et al. 2009) and 2017 (Gerig et al. 2018). We sampled a large variety of meshes from each dataset with different attributes, and formed pairs with ordered attributes. For the meshes within each dataset, we also consistently distributed a good number of landmarks with triplet/quadriplet relations to form the chart-based structure, allowing the generative model to work on 3D meshes. Table 1 summarizes the information of all the datasets. Details on data preparation can be found in the supplementary document.

Implementation

We implemented Rank3DGAN on top of multi-chart 3D GAN (Hamu et al. 2018) with the same hyperparameter setting. The structure of the critic D is modified to adapt for the incorporated ranker as in Figure 2, while the structure of the generator G remains intact. We modelled the encoder E with the same architecture of the vanilla critic D. For the multi-chart representation, we sampled 64×64 images from human and bird flattened meshes, and 128×128 for faces. For the training, we set the hyperparameters $\lambda = 10$, $\nu = 1$, and trained the networks for 300 epochs on all datasets. Similarly to (Hamu et al. 2018), we activated the landmark consistency layer after 50 training epochs for human and bird datasets. This layer is not used for faces since they have a single chart. Finally, we obtained face textures using the nearest neighbors in the real face dataset for better rendering of the results.

Qualitative Results

To compare our method with recent 3D generative model approaches, we extended these models to the conditional setting using semantic attributes, including the voxel 3D GAN (Wu et al. 2016), and AtlasNet (Groueix et al. 2018) generative models. For the former, we have incorporated a ranking network and trained the model similarly to RankCGAN (Saquil, Kim, and Hall 2018) procedure. For the latter, we focused on the task of mesh generation from point clouds input and concatenated the latent variable r to the latent shape representation. We also added the ranking loss to the global objective function, while the ranker is a CNN that takes the point cloud coordinates and produces a ranking score.

We trained these models and Rank3DGAN for 300 epochs using 5000 meshes, 5000 pairwise comparison meshes of the human shape dataset (Pishchulin et al. 2017) and their corresponding point clouds with respect to *weight* attribute. Figure 3 shows the comparison between the selected methods.

Note that the voxel 3D GAN learned the desired transformation, but the results are low in resolution (64^3 dimensions) without surface connectivity and continuity compared to the mesh generated by our method. For AtlasNet interpolation, we remarked that the obesity of human mesh is represented by some vertices getting farther of the mesh. We alleged this behaviour to the objective function being optimal in this scenario where the resulting mesh is a close reconstruction to the input point cloud while the *weight* ranking constraint is satisfied by moving some vertices outside the 3D shape. The interpolation failed to provide the desired output as we do not have access to the ground truth point cloud output given an input point cloud and a specific latent variable r value.

Quantitative Results

We used the metric Fréchet Inception Distance (FID) (Heusel et al. 2017) for evaluating the quality of generated charts. It consists of calculating the Fréchet Distance of two multivariate Gaussians that are estimated from two datasets of real and generated image features. We modelled the feature extraction

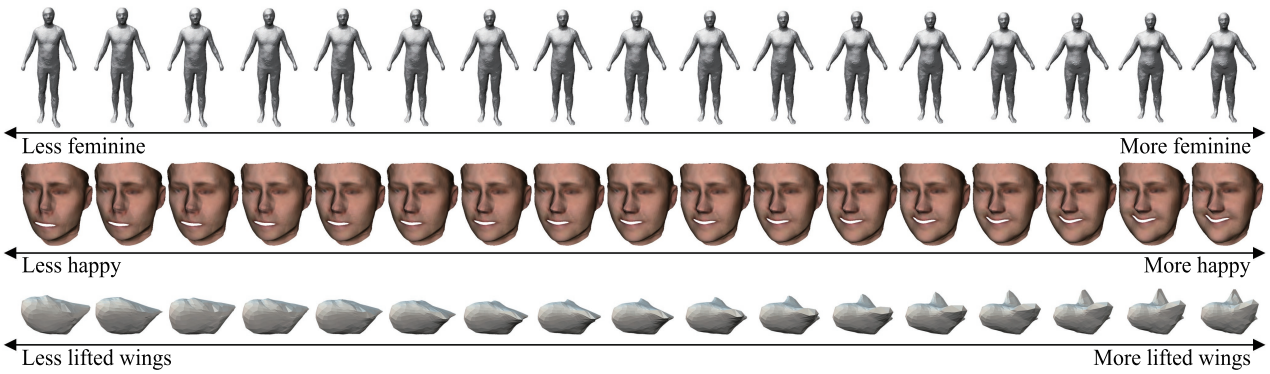

Figure 5: Generated mesh interpolation in the following three datasets: **Top:** human shape, **Middle:** face, **Bottom:** bird with respect to the attributes *weight, happy, lifted wings* respectively.

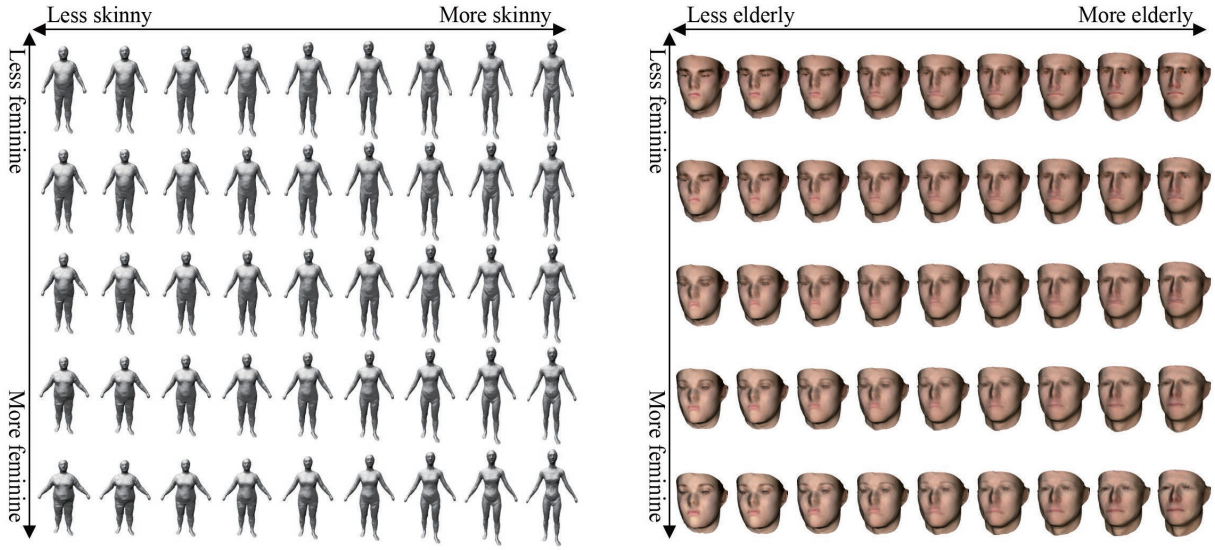

Figure 6: Generated 2D mesh interpolation in the following two datasets: **Left:** human shape with respect to (*weight, gender*) attributes and **Right:** face with respect to (*age,gender*) attributes.

network by FCN (Long, Shelhamer, and Darrell 2014) and calculated FID with respect to each chart in the multi-chart structure, then we averaged the obtained values to get the final mean-FID result.

We trained FCN for segmentation task in face, human, bird datasets. Face dataset (Paysan et al. 2009) provides 4 segment labels along with the face morphable model. We sampled 5000 meshes whose coordinates and labels are represented using the multi-chart structure to create a dataset of $\{(C_i, l_i)\}_{i=1}^{5000|F|}$, with $|F| = 1$ for one quadriplet, $C_i \in \mathbb{R}^{3\times128\times128}$ and $l_i \in [0\ ..\ 3]^{128\times128}$ for the coordinates and label chart. For human (Pishchulin et al. 2017) and bird (Kanazawa et al. 2018) datasets, we manually labelled the mesh vertices and sampled 5000 meshes, resulting in datasets of $\{(C_i, l_i)\}_{i=1}^{5000|F|}$ with $C_i \in \mathbb{R}^{3\times64\times64}$, $|F| = 16$, $l_i \in [0\ ..\ 7]^{64\times64}$ for human meshes, and $|F| = 11$, $l_i \in [0\ ..\ 4]^{64\times64}$ for bird meshes.

Once FCN (Long, Shelhamer, and Darrell 2014) is trained for segmentation task, we extract features of 2048 dimensions from conv7 layer, modified for this purpose, using input real and generated chart images of the same size to build two sets of real and generated features. To investigate the variation in the quality of generated images, we compared the calculated mean-FID of Rank3DGAN with multi-chart 3DGAN (Hamu et al. 2018) at every 15 epochs on all the datasets (see Figure 4). Note that the mean-FID value differences in-between becomes less significant as the models progress through the training epochs, indicating competing performance (with varying attributes) to multi-chart 3DGAN.

We also compared Rank3DGAN against voxel 3D GAN (Wu et al. 2016) and AtlasNet (Groueix et al. 2018) using the volumetric IoU, Chamfer distance and normal consistency metrics on human, face, bird datasets. We selected the attributes *weight*, *happy*, *lifted wings* for the training and estimated the closest generated mesh to the ground-truth mesh by minimizing L_2 loss. Quantitative results are shown in Table

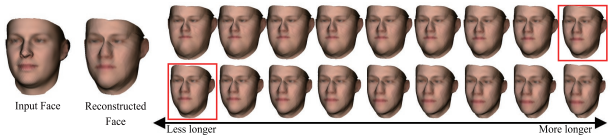

Figure 7: Face mesh editing with respect to *height* attribute.

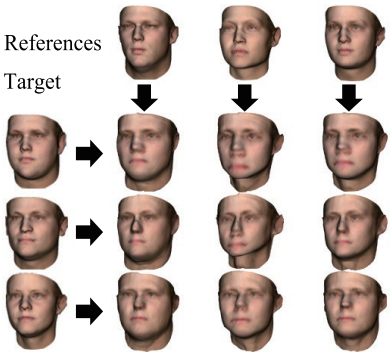

Figure 8: Face attribute transfer according to *weight* attribute.

2. We notice that our method achieves the highest normal consistency and IoU scores as well as a competing Chamfer distance. We note that a mesh is obtained from a voxel using Marching Cube and IoU is not evaluated for AtlasNet and face dataset since the obtained meshes are not watertight.

Applications

Mesh generation: To demonstrate the generative capability of our model, we choose single and double semantic attributes that span an interpolation line and plane respectively, where we fix the latent vector z and vary the latent relative variables in the interval $[-1, 1]$ in order to change the desired relative attributes. Figure 5 shows how the generated meshes vary with respect to the value of the following semantic variables: *weight, lifted wings, happy*. Figure 6 focuses on mesh interpolation with respect to two attributes. We select *weight, gender* for human meshes and *gender, age* for face meshes. We remark that the generated meshes are smooth and the attributes variation is coherent with the selected relative attributes.

Mesh editing: This is an interesting feature for 3D object software tools. It enables the 3D graphic designer to alter high-level attributes in the mesh, that can be customised according to the user intent. This application consists of mapping an input mesh onto the subjective scale by estimating its reconstruction using the latent variables r, z, and then editing the reconstructed mesh by changing the value of $r \in [-1, 1]$. Figure 7 highlights this application in face meshes using *height* attribute. The framed mesh is the reconstructed input mesh with latent variables r^*, z^*, while z^* is fixed, the first and second rows are the interpolation of r in the range $[-1, r^*]$ and $[r^*, 1]$ respectively, denoting generated meshes of subjectively less or more emphasis on the chosen attribute.

Mesh attribute transfer: We can also transfer the subjective strength of an attribute from a reference mesh to a target mesh. Using the encoder E, we quantify the semantic latent variable r of the reference mesh, and then we edit the target mesh with the new estimated semantic value. Figure 8 shows this application for 3D faces in the case of *weight* attribute.

Discussion and Conclusion

We introduce Rank3DGAN, a GAN architecture that synthesises 3D meshes based on their multi-chart structure representation while manipulating local semantic attributes defined relatively using a pairwise comparisons of meshes. We have shown through experimental results that our model is capable of controlling a variety of semantic attributes in generated meshes using a subjective scale. The main limitations of our model are inherited from multi-chart 3D GAN (Hamu et al. 2018), which are the restriction to zero-genus surfaces, *e.g.* sphere-like or disk-like surface and the usage of a fixed template mesh for reconstructing the mesh from the generated charts. Moreover, since the charts represent different parts of the mesh, the pre-processing chart normalization implicitly induces a loss of some global mesh attributes such as the *height* of human shape, which complicates the task of learning a subjective measure using the ranking function. Possible extension of this work can focus on finding a novel way to represent global attributes so that it will not be lost in the mesh processing step. Another line of work can concentrate on enhancing the quality of the generated meshes by generating high resolution charts using other extensions of GANs or using advanced sampling and reconstruction techniques of the 2D flat-torus. Finally, other interesting applications can be derived from this work, such as 3D style transfer and 3D mesh reconstruction from RGB images.

Acknowledgements

We thank Wenbin Li, Christian Richardt, Neill D.F. Campbell for the discussion about the main ideas of the paper. This work is partially funded by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 665992, CDE - the UK's EPSRC Centre for Doctoral Training in Digital Entertainment (EP/L016540/1), CAMERA - the RCUK Centre for the Analysis of Motion, Entertainment Research and Applications (EP/M023281/1), and a gift from Adobe.

References

Achlioptas, P.; Diamanti, O.; Mitliagkas, I.; and Guibas, L. J. 2018. Learning representations and generative models for 3d point clouds. In *ICML*.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein GAN. *CoRR* abs/1701.07875.

Boscaini, D.; Masci, J.; Rodolà, E.; and Bronstein, M. M. 2016. Learning shape correspondence with anisotropic convolutional neural networks. In *NIPS*.

Burges, C. J. C.; Shaked, T.; Renshaw, E.; Lazier, A.; Deeds, M.; Hamilton, N.; and Hullender, G. N. 2005. Learning to rank using gradient descent. In *ICML*.

Chaudhuri, S.; Kalogerakis, E.; Giguere, S.; and Funkhouser, T. A. 2013. Attribit: content creation with semantic attributes. In *UIST*.

Chen, Z., and Zhang, H. 2019. Learning implicit fields for generative shape modeling.

Choy, C. B.; Xu, D.; Gwak, J.; Chen, K.; and Savarese, S. 2016. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV*.

Gerig, T.; Morel-Forster, A.; Blumer, C.; Egger, B.; Lüthi, M.; Schönborn, S.; and Vetter, T. 2018. Morphable face models - an open framework. In *FG*.

Girdhar, R.; Fouhey, D. F.; Rodriguez, M.; and Gupta, A. 2016. Learning a predictable and generative vector representation for objects. In *ECCV*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NIPS*.

Groueix, T.; Fisher, M.; Kim, V. G.; Russell, B. C.; and Aubry, M. 2018. A papier-mâché approach to learning 3d surface generation. In *CVPR*.

Gu, X.; Gortler, S. J.; and Hoppe, H. 2002. Geometry images. *ACM Trans. Graph.*

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C. 2017. Improved training of wasserstein gans. In *NIPS*.

Hamu, H. B.; Maron, H.; Kezurer, I.; Avineri, G.; and Lipman, Y. 2018. Multi-chart generative surface modeling. *ACM Trans. Graph.*

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*.

Joachims, T. 2002. Optimizing search engines using clickthrough data. In *SIGKDD*.

Kalogerakis, E.; Chaudhuri, S.; Koller, D.; and Koltun, V. 2012. A probabilistic model for component-based shape synthesis. *ACM Trans. Graph.*

Kanazawa, A.; Tulsiani, S.; Efros, A. A.; and Malik, J. 2018. Learning category-specific mesh reconstruction from image collections. In *ECCV*.

Kaneko, T.; Hiramatsu, K.; and Kashino, K. 2018. Generative adversarial image synthesis with decision tree latent controller. In *CVPR*.

Liu, J.; Kuipers, B.; and Savarese, S. 2011. Recognizing human actions by attributes. In *CVPR*.

Long, J.; Shelhamer, E.; and Darrell, T. 2014. Fully convolutional networks for semantic segmentation. *CoRR* abs/1411.4038.

Maron, H.; Galun, M.; Aigerman, N.; Trope, M.; Dym, N.; Yumer, E.; Kim, V. G.; and Lipman, Y. 2017. Convolutional neural networks on surfaces via seamless toric covers. *ACM Trans. Graph.*

Masci, J.; Boscaini, D.; Bronstein, M. M.; and Vandergheynst, P. 2015. Geodesic convolutional neural networks on riemannian manifolds. In *ICCV*.

Mescheder, L. M.; Oechsle, M.; Niemeyer, M.; Nowozin, S.; and Geiger, A. 2019. Occupancy networks: Learning 3d reconstruction in function space.

Monti, F.; Boscaini, D.; Masci, J.; Rodolà, E.; Svoboda, J.; and Bronstein, M. M. 2017. Geometric deep learning on graphs and manifolds using mixture model cnns. In *CVPR*.

Odena, A.; Olah, C.; and Shlens, J. 2017. Conditional image synthesis with auxiliary classifier gans. In *ICML*.

Parikh, D., and Grauman, K. 2011. Relative attributes. In *ICCV*.

Paysan, P.; Knothe, R.; Amberg, B.; Romdhani, S.; and Vetter, T. 2009. A 3d face model for pose and illumination invariant face recognition. *AVSS*.

Pishchulin, L.; Wuhrer, S.; Helten, T.; Theobalt, C.; and Schiele, B. 2017. Building statistical shape spaces for 3d human modeling.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*.

Saquil, Y.; Kim, K. I.; and Hall, P. M. 2018. Ranking cgans: Subjective control over semantic image attributes. In *BMVC*.

Sinha, A.; Bai, J.; and Ramani, K. 2016. Deep learning 3d shape surfaces using geometry images. In *ECCV*.

Sinha, A.; Unmesh, A.; Huang, Q.; and Ramani, K. 2017. Surfnet: Generating 3d shape surfaces using deep residual networks. In *CVPR*.

Soltani, A. A.; Huang, H.; Wu, J.; Kulkarni, T. D.; and Tenenbaum, J. B. 2017. Synthesizing 3d shapes via modeling multi-view depth maps and silhouettes with deep generative networks. In *CVPR*.

Souri, Y.; Noury, E.; and Adeli, E. 2016. Deep relative attributes. In *ACCV*.

Streuber, S.; Quiros-Ramirez, M. A.; Hill, M. Q.; Hahn, C. A.; Zuffi, S.; O'Toole, A. J.; and Black, M. J. 2016. Body talk: crowdshaping realistic 3d avatars with words. *ACM Trans. Graph.*

Tan, Q.; Gao, L.; Lai, Y.; and Xia, S. 2018. Variational autoencoders for deforming 3d mesh models. In *CVPR*.

Tao, R.; Smeulders, A. W. M.; and Chang, S. 2015. Attributes and categories for generic instance search from one example. In *CVPR*.

Tompkin, J.; Kim, K. I.; Pfister, H.; and Theobalt, C. 2017. Criteria sliders: Learning continuous database criteria via interactive ranking. In *BMVC*.

Wu, J.; Zhang, C.; Xue, T.; Freeman, B.; and Tenenbaum, J. 2016. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *NIPS*.

Xue, T.; Liu, J.; and Tang, X. 2012. Example-based 3d object reconstruction from line drawings. In *CVPR*.

Yan, X.; Yang, J.; Sohn, K.; and Lee, H. 2016. Attribute2image: Conditional image generation from visual attributes. In *ECCV*.

Yümer, M. E.; Chaudhuri, S.; Hodgins, J. K.; and Kara, L. B. 2015. Semantic shape editing using deformation handles. *ACM Trans. Graph.*

Zhu, J.; Krähenbühl, P.; Shechtman, E.; and Efros, A. A. 2016. Generative visual manipulation on the natural image manifold. In *ECCV*.